



Guía rápida Instalación Selenium Automatización de Pruebas para Aplicaciones Web

Departamento de Auditoria y Calidad de los Sistemas

Junio 2011



INDICE

PRESENTACIÓN	1	DISEÑO Y EJECUCION DE UN SCRIPT DE PRUEBA	11
OBJETIVO	3	PLANTILLA PARA GENERAR SCRIPTS DE PRUEBA	12
REQUERIMIENTOS TÉCNICOS	4		
PROCEDIMIENTO GENERAL PARA LA INSTALACION DE SELENIUM	5		
CONFIGURACION DE SELENIUM IDE	6		
CONFIGURACIÓN DE SELENIUM IDE PARA GENERAR SCRIPTS JAVA	7		
CONFIGURACIÓN DEL SERVIDOR “SELENIUM SERVER”	8		
CONFIGURACIÓN DEL “SELENIUM RC” (AMBIENTE EN ECLIPSE)	9		
DISEÑO Y EJECUCION DE UN SCRIPT DE PRUEBA	10		

PRESENTACIÓN

La automatización de las pruebas consiste en utilizar una herramienta de software para ejecutar pruebas repetidas en contra de la aplicación que sea probada. Esto es una responsabilidad para pruebas de regresión.

Selenium se compone de múltiples herramientas de software, cada una con una función diferente para el apoyo a la automatización de pruebas. Para la automatización de pruebas nos basaremos en el uso de tres herramientas:

- Selenium IDE
- Selenium Remote Control (RC)
- Selenium Server

Sin embargo la suite completa de herramientas de Selenium puede proporcionar diferentes soluciones a problemas con la automatización de pruebas. Una de las características clave de Selenium es el apoyo para realizar pruebas sobre varios exploradores.

Selenium IDE

Es un entorno integrado de desarrollo (Integrated Development Environment) para la creación de scripts de pruebas de Selenium. Es un complemento implementado como una extensión del explorador Mozilla Firefox y permite la grabación, edición, depuración de errores (debug) de los scripts. Selenium IDE incluye el núcleo completo de Selenium permitiendo de una manera rápida y sencilla la grabación y ejecución de pruebas en el ambiente en el cual se esté ejecutando.

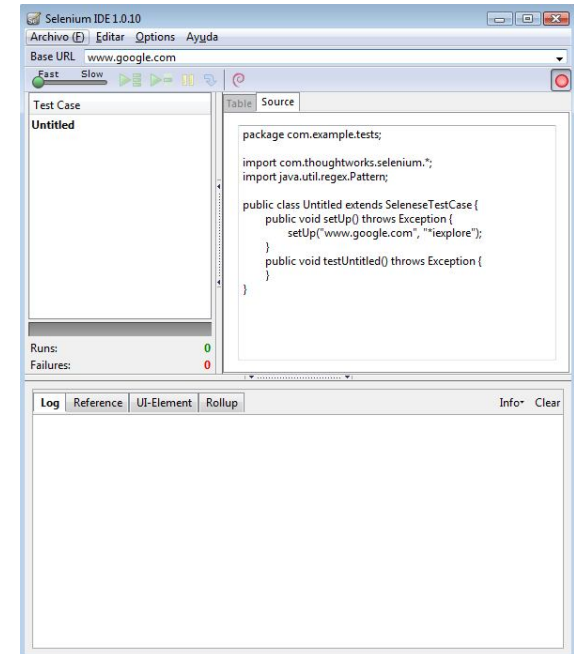
Selenium IDE no solo es una herramienta para grabar scripts de prueba. Sino también incorpora la función de editar los scripts a “mano” y permite seleccionar dentro de diferentes tipos de lenguaje de programación para mostrar los scripts grabados.

Ventajas:

- Grabación fácil de scripts de prueba.
- Selección inteligente de elementos HTML.
- Auto-completar comandos comunes de Selenium
- Depuración de errores
- Guardar scripts de prueba en lenguaje HTML, Ruby, Java entre otros formatos.

Desventajas

- El complemento solo puede ser instalado sobre Mozilla Firefox.



Selenium Remote Control (RC)

Es una herramienta para la automatización de pruebas que permite escribir y ejecutar pruebas automatizadas en diferentes lenguajes de programación (Java, Javascript, Ruby, HP, Python, Perl and C#) y soporta casi cualquier browser del mercado (Internet Explorer, Firefox, Chrome, Safari, Opera, etc.)

Selenium RC se compone de dos partes:

- Un servidor que funciona como un Proxy que recibe las sentencias y las transforma en lenguaje HTTP para el explorador.
- Las librerías de Selenium para el lenguaje preferido.

Selenium Server

El servidor de Selenium recibe comandos del script de pruebas, los interpreta e informa a su programa los resultados de ejecutar las pruebas.

El servidor recibe los comandos Selenese del script de pruebas utilizando simple HTTP con peticiones GET / POST. Esto significa que se puede utilizar cualquier lenguaje de programación que pueda enviar peticiones HTTP para automatizar las pruebas de Selenium en el navegador.



Objetivo

Esta guía de usuario introduce al uso de Selenium y proporciona una serie de pasos para lograr una instalación exitosa de las herramientas y la configuración y ejecución de un script de ejemplo.

REQUERIMIENTOS TÉCNICOS

Se tiene contemplado la operación de las herramientas de **Selenium** en los equipos de cómputo del Departamento de Auditoria y Calidad de los Sistemas del Instituto Federal Electoral; para tal fin es necesario que el equipo de cómputo donde se operen las herramientas, cuente con los siguientes requisitos técnicos:

Hardware

- Procesador Intel(R) Core(TM)2 ó superior.
- 1.0 GB de memoria RAM (mínima recomendada).
- 1 GB de espacio libre en disco duro (mínimo recomendado).

Software

- Sistema operativo: Windows Vista/XP®.
- Internet Explorer 7.0 o superior.
- Mozilla Firefox 3.6.x (La herramienta Selenium IDE **No funciona** actualmente con la versión 4.0 de FireFox)
- Eclipse SDK Version: 3.6.2

PROCEDIMIENTO GENERAL PARA LA INSTALACION DE SELENIUM

A continuacion se muestra un panorama general para la instalacion y ejecucion de un script de pruebas en Selenium



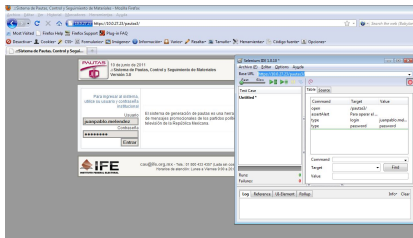
¿Cómo funciona?

Siguiendo estos sencillos pasos

PASO 1

Ejecutamos Selenium IDE

Se abre un explorador Firefox y graba el modulo a automatizar por medio de la herramienta Selenium IDE



PASO 2

Levantamos el Servidor

Abrimos la línea de comandos de Windows. Y ejecutamos el comando:

```
C:\>java -jar selenium-server-2.0.jar
```

PASO 3

Levantamos el ambiente Selenium RC en Java.

Se accede a eclipse:

C:\eclipse\eclipse.exe

Se da un nombre a un Nuevo Proyecto, así como al Test Case.

1. Nuevo proyecto:

File→New→Other→Java Project

2. Crear TestCase:

File→New→Other→Java→JUnit→JUnit Test Case

3. Se agregan los JARs de Selenium al proyecto.

PASO 4

Se ejecuta el script de pruebas

Se copia el script generado con la herramienta Selenium IDE y se pega en el Test Case que acabamos de crear en Java.

Ejecutar el script de Prueba Ctrl+F11,

Observar el resultado de la automatización.



CONFIGURACION DE SELENIUM IDE

A continuación se sigue el siguiente procedimiento para la configuración de la herramienta Selenium IDE en Mozilla Firefox.



¿Cómo se configura?

Siguiendo estos sencillos pasos.

PASO 1

Se abre un explorador Firefox y se ingresa a la dirección:

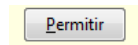
<http://release.seleniumhq.org/selenium-ide/1.0.10/>



Se hace clic en link [1.0.10](#) del apartado **Selenium IDE**

PASO 2

Se mostrara una barra de alerta en la parte superior de la ventana. Se da clic en el botón **Permitir**



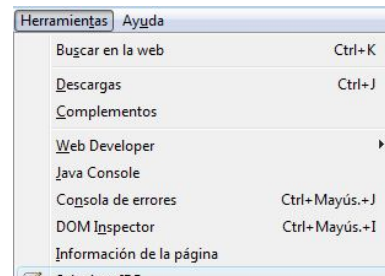
En la ventana de instalación se da clic en la opción **Instalar ahora**



PASO 3

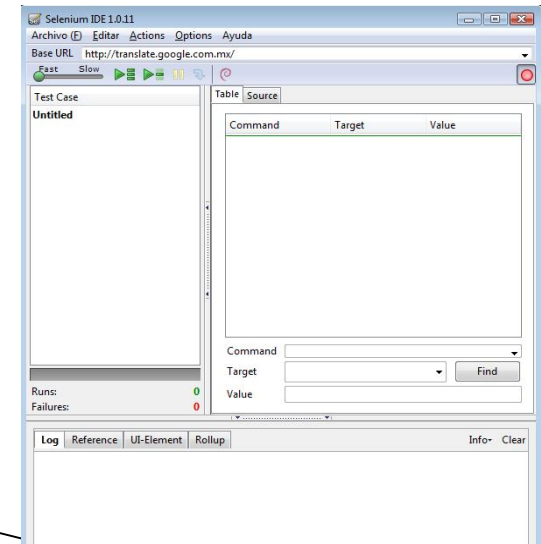
Una vez Instalado el complemento se reinicia Firefox para que los cambios tengan efecto.

Una vez reiniciado se puede ejecutar la herramienta desde: **Herramientas→Selenium IDE**



PASO 4

Se ejecutara la herramienta Selenium IDE y se mostrara la siguiente interfaz:



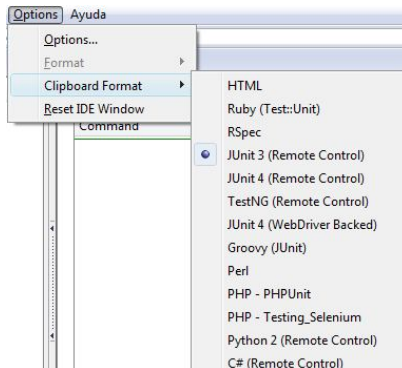
Configuración de Selenium IDE para generar scripts Java

Una vez abierta la herramienta Selenium IDE es necesario configurarla para que los scripts de prueba que grabemos se muestren en lenguaje Java.

PASO 1

Dentro de la interfaz de Selenium IDE configuramos el lenguaje en el que se mostrara el script seleccionando la opción JUnit3:

Options→Clipboard Format→JUnit3 (Remote Control)



PASO 2

Ingresamos la URL Base que sera la direccion a la cual realizaremos nuestra prueba:

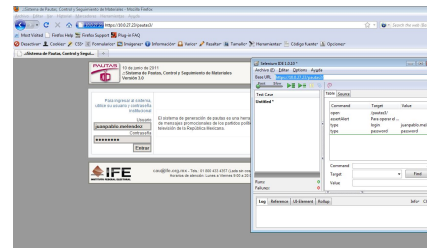
Base URL `https://10.0.27.23/pautas3/`

Para comenzar a grabar nuestra prueba activamos nuestro botón de grabación:



PASO 3

Sin cerrar la herramienta Selenium IDE Comenzamos a ejecutar dentro de la ventana abierta del explorador Firefox el flujo el cual queremos automatizar.



PASO 4

Al terminar el flujo deseado, regresamos a la herramienta y detenemos la grabacion.



Por último copiaremos el código que se genero dentro de la pestaña **Source** y lo pegaremos en nuestra plantilla en Eclipse.

```
package com.example.tests;

import com.thoughtworks.selenium.*;
import java.util.regex.Pattern;

public class Untitled extends SeleniumTestCase {
    public void setUp() throws Exception {
        setUp("https://10.0.27.23/pautas3/", "chrome");
    }

    public void testUntitled() throws Exception {
        selenium.open("/pautas3/");
        assertEquals("Para operar el sistema se requiere Internet Explorer 7 o superior", selenium.getAlert());
        selenium.type("login", "juampablo.melendez");
        selenium.type("password", "password");
        selenium.type("login", "mandando.callado");
        selenium.click("submit");
        selenium.waitForPageToLoad("30000");
        selenium.click("links:Registrar grupos");
        selenium.waitForPageToLoad("30000");
        selenium.click("vigente");
        selenium.type("denominacion", "pablo");
        selenium.select("idEstado", "label=AGUASCALIENTES");
        selenium.select("idEstado", "label=DISTRITO FEDERAL");
        selenium.click("option[@value=9]");
        selenium.select("idMunicipio", "label=005 - GUSTAVO A. MADERO");
        selenium.type("domicilio", "acoya");
    }
}
```

Nota: El código a copiar es el que se encuentra dentro del método:

```
public void testUntitled() throws
Exception {

}
```

Configuración del Servidor “Selenium Server”

Antes de correr nuestro script que generó la herramienta Selenium IDE debemos instalar el Selenium Server Standalone.



¿Cómo se configura?

Siguiendo estos sencillos pasos.

PASO 1

Descargamos el archivo

Selenium-Server-2.0.jar

Y lo depositamos dentro de una carpeta con el nombre de "SeleniumServer" dentro de la carpeta raíz del sistema. (Ej. C:\SeleniumServer).



PASO 2

Ejecutamos una línea de comando (Con privilegios de Administrador para el caso de widows VISTA).



Símbolo del sistema

PASO 3

Dentro de la línea de comandos ingresamos a la carpeta donde se encuentra el archivo selenium-server-2.0.jar con el comando:

cd [Ruta completa de la carpeta]
(Ej. cd C:\SeleniumServer)

```
C:\Users\life>cd C:\SeleniumServer  
C:\SeleniumServer>_
```

PASO 4

Por ultimo ejecutamos el comando :

java -jar selenium-server-2.0.jar
para levantar el servidor

```
C:\SeleniumServer>java -jar selenium-server-2.0.jar  
18:11:03.217 INFO - java: Sun Microsystems Inc. 20.0-b11  
18:11:03.217 INFO - OS: Windows Vista 6.0 x86  
18:11:03.217 INFO - v2.0 [b3], with Core v2.0 [b3]  
18:11:03.514 INFO - RemoteWebDriver instances should connect to: http://127.0.0.1:4444/wd/hub  
18:11:03.514 INFO - Version: Jetty/5.1.x  
18:11:03.529 INFO - Started HttpContext[/selenium-server/driver/,selenium-server-2.0.jar]  
18:11:03.529 INFO - Started HttpContext[/,/]  
18:11:03.592 INFO - Started org.openqa.jetty.jetty.servlet.ServletHandler  
18:11:03.592 INFO - Started HttpContext[/wd/,wd]  
18:11:03.607 INFO - Started SocketListener on 0.0.0.0:4444  
18:11:03.607 INFO - Started org.openqa.jetty.jetty.Server@1c184f4
```

Nota: Si se cierra la ventana del símbolo del sistema el servidor **se detiene**.

Configuración del “Selenium RC” (Ambiente en Eclipse)

A continuación se sigue el siguiente procedimiento para la configuración del ambiente para utilizar Selenium correctamente.



¿Cómo se configura?

Siguiendo estos sencillos pasos.

PASO 1

Se descarga y se descomprime el archivo eclipse-SDK-3.6.2-win32.zip en la unidad C:\



Importante: Se recomienda que se descomprima en la ruta mencionada.

El eclipse se encuentra en la ruta:
(DIS-UNICOM/Subdirección de Análisis y Diseño de Sistemas/Departamento de Calidad de Sistemas/01 Oportunidades de mejora del DACS/Software_Watij)

PASO 2

Se copia la carpeta Librerías (selenium-java-2.0b3) en C:\eclipse, esta carpeta contiene las librerías esenciales para su funcionamiento.



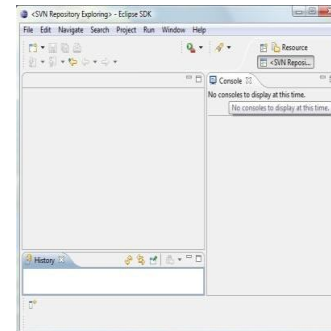
En caso de que tenga en otra ubicación eclipse copie la carpeta librerías en ella.

PASO 3

Se ejecuta eclipse desde la siguiente ruta:

C:\eclipse\eclipse.exe

De esta manera se configura el ambiente para utilizar Selenium en Eclipse.



Importante:

Los archivos a descargar:

- **Selenium-Server-2.0.jar**
- **selenium-java-2.0b3.zip**

Se encuentran en la siguiente ruta del subversión:

(DIS-UNICOM/Subdirección de Análisis y Diseño de Sistemas/Departamento de Calidad de Sistemas/01 Oportunidades de mejora del DACS/Software_Selenium), o bien, solicitar al jefe del DACS el paquete de Software que se requiere para la configuración del ambiente de Selenium.

DISEÑO Y EJECUCION DE UN SCRIPT DE PRUEBA

Después de haber configurado el ambiente, ahora se puede diseñar y ejecutar scripts de prueba para comprobar el ambiente.

Para ejecutar esta prueba, Se realizará una búsqueda de la palabra IFE y posteriormente e ingrese a ella en el buscador Google.



¿Cómo se diseña?

Siguiendo estos sencillos pasos.

PASO 1

Una vez abierto eclipse, se crea un nuevo proyecto a través de la siguiente secuencia del menú:

File→New→Other→Java Project

1. Se ingresa en el campo **Project name** el nombre de nuestro proyecto ejemplo: **EjemploSelenium**, posteriormente se hace clic en el botón **Finish**.

2. Se desplegará una ventana llamada **Open Associated Perspective?**. Se hace clic en **Yes**.

De esta forma se crea la estructura del proyecto en general.

PASO 2

A continuación se da el nombre al script en TestCase a través de la siguiente secuencia del menú:

File→New→Other→Java→JUnit→JUnit Test Case

1. Se ingresa en el campo **Name** el nombre del script ejemplo:

TestBuscarPalabra,

2. En el campo **Package** se utilizara la siguiente nomenclatura.

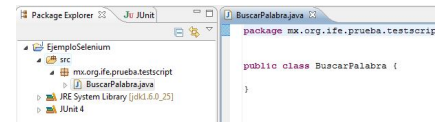
mx.org.ife.<Nombre corto o siglas del sistema>.testscript

Posteriormente se da clic en **Finish**

3. Se muestra una ventana llamada **New JUnit Test Case**, haga clic en **OK**, para agregar las **librerías JUnit**.

PASO 3

Se observa en eclipse el nombre del proyecto y nombre del script creado.



PASO 4

Finalmente se copian todos los **JARs** de la carpeta **selenium-java-2.0b3** al proyecto.

1. Se hace clic con el botón derecho del ratón sobre el nombre del proyecto y seleccione **Properties**.
2. De la pantalla **Properties**, se hace clic en:

Java Build Path →Libraries → Add External JARs

3. Seleccione de la siguiente ruta todo los JARs **C:\eclipse\selenium-java-2.0b3** y haga clic en el botón **Abrir**.

4. Finalmente haga clic en **OK**.

DISEÑO Y EJECUCION DE UN SCRIPT DE PRUEBA

Después de haber configurado el ambiente, ahora se puede diseñar y ejecutar scripts de prueba para comprobar el ambiente.

Para ejecutar esta prueba, Se realizará una búsqueda de la palabra IFE y posteriormente e ingrese a ella en el buscador Google.



¿Cómo se ejecuta?

Siguiendo estos sencillos pasos.

PASO 1

Grabar nuestro flujo con Selenium IDE

Copiaremos el código que se genere dentro de la pestaña **Source** y lo pegaremos en nuestra plantilla en Eclipse.

```
package com.sample.test;

import com.thoughtworks.selenium.*;
import java.util.regex.Pattern;

public class Untitled1 extends SeleniumTestCase {
    public void setUp() throws Exception {
        setUp("https://0.0.0.0:22/pautas/", "chrome");
    }

    public void testUntitled1() throws Exception {
        selenium.open("https://0.0.0.0:22/pautas/");
        selenium.waitForPageToLoad(30000);
        selenium.type("login", "jcampablo.melendez");
        selenium.type("password", "jcampablo");
        selenium.click("button");
        selenium.waitForPageToLoad(30000);
        selenium.click("link=Registrar grupos");
        selenium.waitForPageToLoad(30000);
        selenium.click("link=Registrar");
        selenium.type("denominacion", "PABLO");
        selenium.select("estado", "label=AGUASCALIENTES");
        selenium.select("id=estado", "label=INSTITUTO FEDERAL");
        selenium.click("input=Registrar");
        selenium.select("id=Autosugiero", "label=003 - GUSTAVO A. MADERO");
        selenium.type("dominio", "acopa");
        selenium.type("correo", "pablo.melendez@hotmail.com");
        selenium.type("telefono", "74654654");
        selenium.click("id=Registrar");
        selenium.type("nombre", "pablo");
        selenium.waitForPageToLoad(30000);
    }
}
```

PASO 2

La plantilla se modifica dependiendo de los nombres de los paquetes y clases que se le hayan asignado al proyecto.

Nota:

La plantilla se muestra abajo.

PASO 3

Para ejecutar el programa capturado realice cualquiera de los dos procedimientos:

1. El juego de teclas **Ctrl+F11**

O bien la secuencia del menú

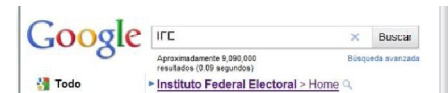
2. **Run→Run**

Nota:

Es necesario hacer ajustes al código generado por la herramienta Selenium IDE ya que el flujo de los sistemas puede variar entre los diferentes exploradores.

PASO 4

Minimice el Eclipse y Observe el resultado de la automatización creada:



Plantilla para generar Scripts de prueba

```
/**
 * Plantilla para Generar scripts de pruebas en Selenium
 */

package mx.org.ife.pruebas.testscript; //Modificar el nombre del paquete dependiendo de su configuración.

/**
 * @author Juan Pablo Melendez Vieyra
 * Departamento de Auditoria y Calidad de los Sistemas
 * Instituto Federal Electoral
 */

import com.thoughtworks.selenium.DefaultSelenium;
import com.thoughtworks.selenium.Selenium;
import junit.framework.TestCase;

public class PlantillaSelenium extends TestCase { //Modificar el nombre de la clase dependiendo de su configuración

    private Selenium selenium;
    protected void setUp() {
        selenium = new DefaultSelenium("localhost", 4444, "*iexplore", "http://www.ife.org.mx/");
//Modificar la URL dependiendo del sistema a automatizar
        selenium.start();
    }
    public void test() {

        /**
         *
         * Copiar AQUI el codigo generado con la herramienta Selenium IDE
         *
         */

    }

    protected void tearDown(){
        selenium.stop();
    }
}
```